Project: Bank Marketing (Campaign)

Week 9 deliverables

Group name: Fleifel-solo

Name: Rania Tarek Fleifel

Email: raniatarekfleifel@gmail.com

Country: Egypt

College: Cairo university Faculty of engineering

Specialization: Data science

Internship Batch: LISUM13: 30

Submission date: 30th November 2022

Data Cleansing and Transformation

After performing raw data exploration in the previous deliverable. It's time we address the issues we found during our analysis.

Factors handled/taken in consideration:

a) Missing data:

Although there are no NaNs in our data, there are ambiguous values such as *unknown*,*other* in the features: *education*,*poutcome*,*contact*. We attempt to look into each to decide on the plan of action

b) High cardinality:

Although this is usually used for categorical features. This problem appears in *job* feature. Using this kind of data to train a predictive model results in non-conclusive models that fail to generalize well. The same issue is caused having numerical features with high variance (e.g. *balance*, *duration*)

c) Nominal & ordinal nature of data:

According to this information, the type of encoding chosen differs

d) High dimensionality:

Mindful of making the features impossible for predictive models to use, we attempt to remove features that are well-represented otherwise

e) Outliers:

To handle outliers, percentile can be used to pinpoint outliers. Lower outliers are considered below the 0.1 percentile and higher outliers are beyond the 0.99 percentile. You can choose to clip the features before the 0.1 percentile and after the 0.99 percentile or cap the features to these values. You could also opt to not drop outliers at all. This depends on information gain which will be discussed in more details in EDA deliverable.

f) Skewed-data

In this project, we attempt to transform all features into numerical features revolving around 0 and 1. This inherently removes skewness. This is done through binning and encoding of features. Other methods used to handle this are scaling features, and using log-transformation are also common to approach normally-distributed features. The latter methods will be applied as needed during EDA to factor for feature importance to the target.

g) Class imbalance

The class imbalance requires avoiding using statistics that use positives and negatives together like accuracy. Although ROC curve is widely used in these cases, it's not preferred when there is skewness in the data because any small change in predictions cause a huge change in the curve. We will use precision and recall to evaluate our models. We also make sure to stratify data between testing and training .

Feature engineering and data manipulation:

Data cleaning:

- Remove whitespaces from strings (strings=columns | column_headers)
- Handle special characters and spaces within strings
- Lower case all strings
- Ensure all columns have synchronous type of data

Imputation:

- Given that the data is ordered, we can deduce the year of each instance.
- Social and economic indicators from Portugal in the time span between May 2008 and Dec 2010 are quite important in our problem space; Given that these were considered global recession years. We use data available at data.nasdaq and bpstat.bportugal to represent employment, consumer price index, consumer confidence index and Euribor.

Data transformation:

1) Poutcome

- *poutcome=unknown* & (*pdays==999* | *previous=0*) → *poutcome=non_existant*
- *poutcome=unknown* & *pdays!=999* are only 5 entries → drop the rows
- *poutcome_missing* is a binary flag that tells whether *poutcome=other*
- encode *poutcome* into *0 (other, failure)* and *1 (success)*

2) *Education*

- *primary*,*secondary*,*teritary* has a ordinal nature to it, so we use judgment encoding which is the same as label encoding but with contextual logic to the encoding
- sklearn labelencoding encodes classes according to .unique which misses the ordinal nature of the classes, so we opt to specify the classes are encoded as follows: {*unkown:0, primary:1,secondary:2,teritary:3}*
- *education_missing* is a binary flag that tells whether *education=unknown*

3) *Contact*

- *contact_missing* is a binary flag that tells whether *contact=other*
- encode *contact* into *0 (other, failure)* and *1 (success)*

4) *Job*

- 12 unique values will cause high cardinality if left as is and high dimensionality if hot-encoded
- maintain *unknown* as a class-label
- *job_missing* is a binary flag that tells whether *job=unknown*
- Encode values based on value counts w.r.t y (aka frequency encoding).
- This method of encoding is derived from target encoding, with an important difference that only the training y is taken in consideration to deduce the new values of *job*
- This means we can expect over-fitting in training but normal performance in the test.

5) *Day, Year, Month*

- Although numeric, will illude to favoring higher values over lower values which is not correct.
- Deduce *day_of_week* then hot encode it
- Hot encode *year,month*
- Remove *day* feature

6) *Marital*

- Has no ordinal nature so can't label encode, hence hot-encoding is used

7) <u>Balance</u>

- Negative values indicate costumers who owe the bank money and their balance doesn’t cover it
- *Overdraft* is a binary indicator for such costumers
- All negative values are replaced with 0
- Binning the balance into 5 equal categories then use their ordinal nature to label encode them ** the min and max of data is obtained based on training slpit

8) <u>loan,housing,default</u>

- Binary encode these values

9) *<u>Numeric features: age,duration,pdays,campaign,previous,</u>*

- Before normalizing, scaling or using log transformation or even removing outliers, cross-correlation with target and features importance should be discussed first. So for now, leave as is and discuss this further in the EDA deliverable (week10)

10) *<u>Imputed features: 'euribor3m', 'consum_prices_rate', 'consum_conf_ind', 'employed', 'unemployed', 'unemployed_rate'</u>*

- The 0 values in *euribor3m* is intentional, since this value has 3decimal places approximation in its definition
- Same as the previous bullet, heavily dependent on EDA. Will be discussed further in the EDA deliverable (week10)